一、PC寄存器-也叫做程序计数器的介绍（Program Counter Register）（相当于游标或者迭代器）

（[java JVM内存结构之PC寄存器\_clinginh的博客-CSDN博客](https://blog.csdn.net/clinginh/article/details/115364617)）

1、JVM中的程序计数器中，Resgister的命名源于CPU的寄存器，寄存器存储指令相关的现场信息。CPU只有把数据装载到寄存器才能够运行。

2、**注意！JVM中的PC寄存器，并非广义上所指的物理寄存器**，或许将其翻译为PC计数器（或指令计数器、程序钩子、行号指示器）会更加贴切。并且也不容易引起一些不必要的误会。**JVM中的PC寄存器是对物理PC寄存器的一种抽象模拟。是软件层面的概念。**

二、PC寄存器的作用：

１、PC寄存器用来存储指向下条指令的地址，也就是即将要执行的代码。由执行引擎读取下一条指令。（一个栈帧相当于一个方法）（这图表述的不够好，因为**方法模板都是放在方法区中的，所以PC指向的行号肯定是指向方法区中的地址。而这些方法针对于每个线程产生的变量的值应该放入相应的栈帧中，当执行引擎根据PC寄存器的指引进行代码执行时，将产生的结果放入其中或是取其中的值**。）



三、PC特点：

１、它是一块很小的内存空间，几乎可以忽略不记。也就是运行速度最快的存储区域。

２、**在JVM规范中，每个线程都有它自己的程序计数器，是线程私有的，生命周期与线程的生命周期保持一致**。

３、**任何时间一个线程都只有一个方法在执行**，也就是所谓的**当前方法－当前栈帧**。程序计数器会存储当前线程正在执行的Java方法的JVM指令地址；或者，如果是在执行Native方法，则是未指定值（ｕｎｄｅｆｎｅｄ）。

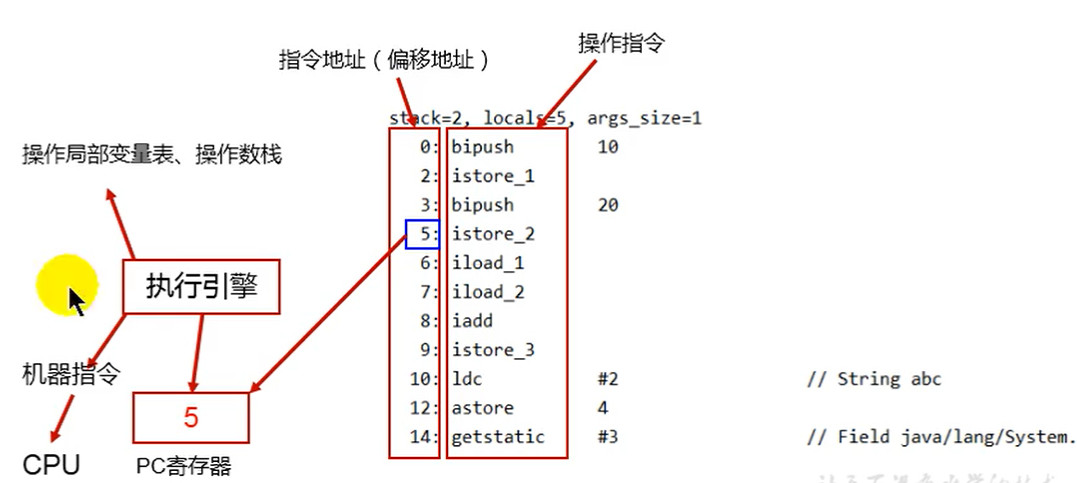
４、PC寄存器是程序控制流的指示器，分支、循环、跳转、异常处理、线程恢复等基础功能都需要依赖这个计数器来完成。

５、字节码解释器工作时就是通过改变这个计数器的值来选取下一条需要执行的字节码指令。

６、PC寄存器是唯一一个在Java虚拟机规范中没有规定任何ＯｕｔＯｔＭｅｍｏｒｙＥｒｒｏｒ（OOM内存溢出异常）情况的区域。

７、方法区、堆区存在GC（垃圾回收），虚拟机栈、本地方法栈与PCR不存在GC，除了PCR，其他区域均可能报OOM异常。

四、例子：



五、常见问题：

1、使用PC寄存器存储字节码指令地址有什么用？为什么使用PC寄存器记录当前线程的执行地址？

答：

1.因为COU需要不同的切换各个线程，这时候切换回来以后，就得知道接着从哪里开始继续执行。

2.JVM的字节码解释器就需要通过改变PC寄存器的值来明确下一条应该执行什么样的字节码指令。

2、PC寄存器为什么会被设定为线程私有？

答：

（并发、并行、串行

[串行，并发与并行\_知识在于积累-CSDN博客\_串行和并行](https://blog.csdn.net/u014489699/article/details/89893438)）

CPU执行策略：分时间片，抢占式执行策略。

CPU时间片即CPU分配给各个程序地时间，每个线程被分配一个时间段，称作它地时间片。

例：

在宏观上:我们可以同时打开多个应用程序，每个程序并行不悖，同时运行。

在微观上：由于只有一个CPU，一次只能处理程序要求地一部分，如何处理公平，一种方法就是引入时间片，每个程序轮流执行。

1.所谓多线程，是宏观并行，微观并发（同一时刻多个线程竞争CPU执行权，但CPU只能执行一个，也就是多个线程不停切换），CPU会不同地做任务切换，这样必然导致经常中断或恢复，如何保证分毫无差？**为了准确记录各个线程正在执行地当前字节码指令地址，最好地办法自然是为每一个线程分配一个PC寄存器，这样各个线程之间便能进行独立计算，从而不会出现相互干扰地情况。**

2.由于CPU时间片限制，众多线程在并发执行过程中，任何一个确定地时刻，一个处理器或者多核处理器中地一个内核，只会执行某个线程中地一条指令。

3.这样必然导致经常中断或恢复，如何保证分毫无差呢？每个线程在创建后，都会产生自己地程序计数器和栈帧，程序计数器在各个线程之间互不影响。